

Helló!

Ez a TEXLAB program rövid leírása.

TEXLAB fájlból először tex fájlt csinálni a texlab-compile paranccsal, aztán lehet futtatni pdflatex-szel így:

```
ruby "...../texlab-compile.rb" "....a-te-fájlod.texlab"
```

```
pdflatex ".....a-te-fájlod.tex"
```

Ez, amit olvasol, egy TEXLAB fájl. A TEXLAB fájlban Ruby és LaTeX kódok váltogatják egymást. LaTeX kóddal kezdődik, és Ruby módba a kisebb-százalék jelekkel lehet átmenni, illetve a százalék-nagyobb jelekkel visszamenni.

A százalékjelek között lehet ruby parancsokat adni. Az egyik ilyen parancs a documentHeader, amit a fájl elején látsz. Ez elkészíti a LaTeX fájl elejére írandó sorokat A paraméterei:

usePackages: milyen csomagokat használjunk. A babel csomag a nyelvi speciális dolgokat csinálja. Az inputenc-nek kell megadni, hogy milyen formátumú a fájl. Ezeken kívül is jónéhány csomagot alaptól betölt.

twocolumn: ha true, akkor kéthasábos lesz a dokumentum

fontsize: "12pt", "11pt" vagy "10pt" (default). a fontméret.

landscape: true esetén a dokumentum fekvő lapon lesz.

A másik parancs a puts, ami kiír egy kifejezést:

Helló

A sor elején lévő százalékjel is átvisz ruby módba.

.)

Ha LaTeX megjegyzést szeretnél írni, akkor sajnos dupla százalékjelet kell írni:

A harmadik lehetőség, hogy ha egy Ruby kifejezés értékét szeretnéd betenni a fájlba, akkor a kisebb-százalék-egyenlő jellel kell nyitni:

Helló Ez lényegében ugyanolyan, mint a puts.

A Rubyban lehet matematikai kifejezéseket írni. Pl. öt meg öt az 10. A hatványozás \*\*, a tizedestörtökben pont van.

Tudsz definiálni változókat:

Most 58 darab almánk van.

A konstansok nagybetűvel kezdődnek. Két konstansunk is van, 2.718281828459045 és 3.141592653589793. Te is definiálhatsz konstans:

A számokra meg lehet adni a pm (pluszminusz) metódussal hibahatárt: 12.5000. Ilyenkor kézzel kell konvertálni Latex képletté a to\_latex metódussal, különben az eredeti szám jelenik meg. Ha puts-t használ, akkor nem kell, mert automatikusan megcsinálja. 13.50000

Relatív is meg lehet adni a hibát 0.50 Szép exponenciális alakokat tud csinálni:  $1.20 \cdot 10^{-10}$ .

Vannak függvényeink. 90 fok az 1.5707963267948966 radián, ennek a szinusza 1.0. Ha egyértelmű, hogy mit akarsz, akkor nem muszáj kiírni a zárójeleket, hanem írhatod pl., hogy 1.0. De ha erre nagyon rászoksz, még sok gondot fog okozni :).

Te is definiálhatsz függvényeket:  $\cosh 4 = 27.308232836016483$  De ez a függvény már amúgy is létezik.

Az adatokat néha érdemes listában tárolni: [1, 2, 3] Ezeken végig lehet menni kétféleképpen is:

1 alma 2 alma 3 alma

vagy

1 alma 2 alma 3 alma .

Az adatokat néha érdemes hash-ben tárolni:

A test tömege 9.1, térfogata 8.1, sűrűsége 1.123456790123457.

A program tud számolni hibaterjedést, úgyhogy érdemes megadni minden adatban a hibahatárt, hogy az eredmény szebb legyen:

A test tömege 9.1, térfogata 8.10, sűrűsége 1.12.

Táblázatokat lehet csinálni.

	oszlopcím
sorcím	érték

1. táblázat. cím

Lehet több oszlopot és sort is. A do..end helyett lehet kapcsos zárójelet használni

Lehet több szintű oszlopokat is, de többszintű sorokat (még) nem.

Figyeld meg, hogy a .to\_latex automatikusan megtörténik, tehát itt sem kell kiírni, akárcsak a puts-nál. Azt is figyeld meg, hogy mindig table, row, column a sorrend az egymásba ágyazásnál.

Érdemes az adatokat ciklusból táblázatba konvertálni:

A rowTitle opcióval lehet megadni, hogy mi legyen a sarokban. Így lehet csinálni szorzótáblát:

	nagy	kicsi
okos	kutya	macska
buta	bálna	egér

2. táblázat. állatok

	tömeg	
	előtte	utána
test1	90	$7.5 \cdot 10^{-4}$

3. táblázat. adatok

	kétszerese
1	2
2	4
3	6

4. táblázat. számolás

x	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

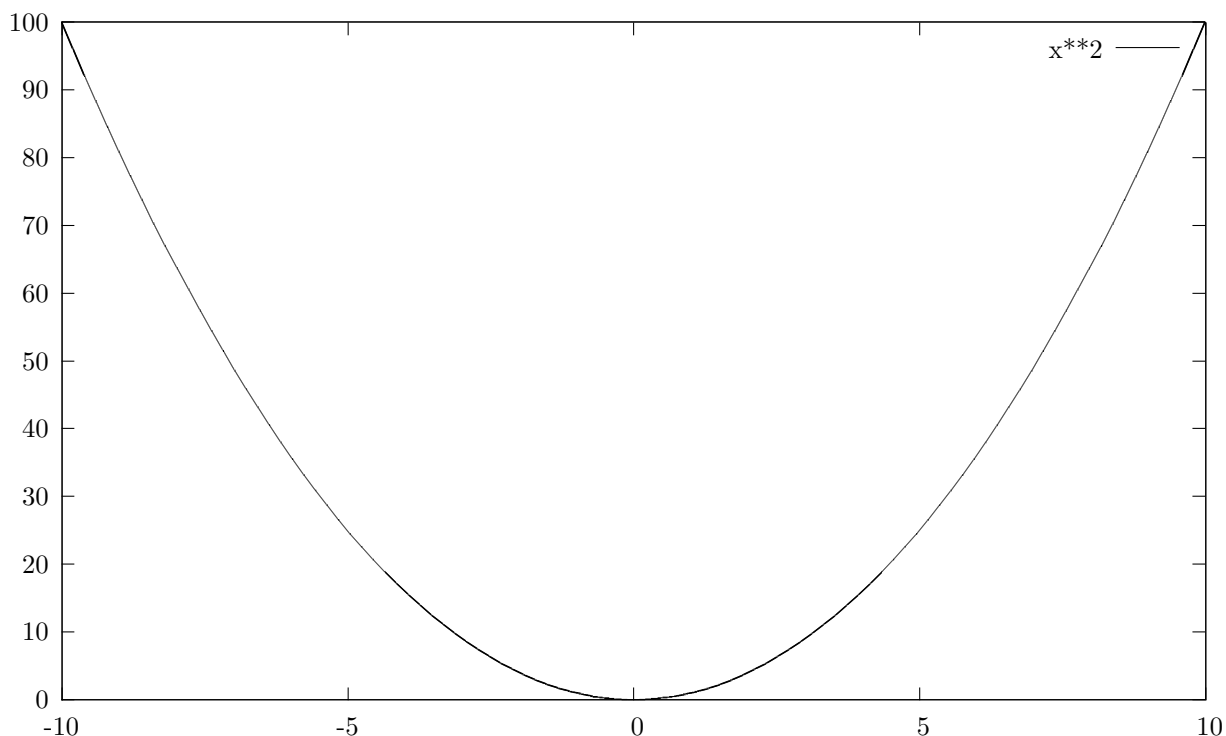
5. táblázat. Szorzótábla

A lehetséges beállítások:  
 placement: "H" - legyen a táblázat pont itt, és ne külön lapon.  
 rowTitle: a sarokban mi legyen  
 a többit lásd <http://rubydoc.info/gems/latex/LatexFile:table>  
 Összetett táblázatok:

(a) Első		(b) Második	
	valami		semmi
h	k	x	o

6. táblázat. Táblák

Grafikonok:



1. ábra.  $x^2$  függvény

Adatpontok:

Illesztés:  $5x + -3$

Illesztett egyenes a grafikonon:

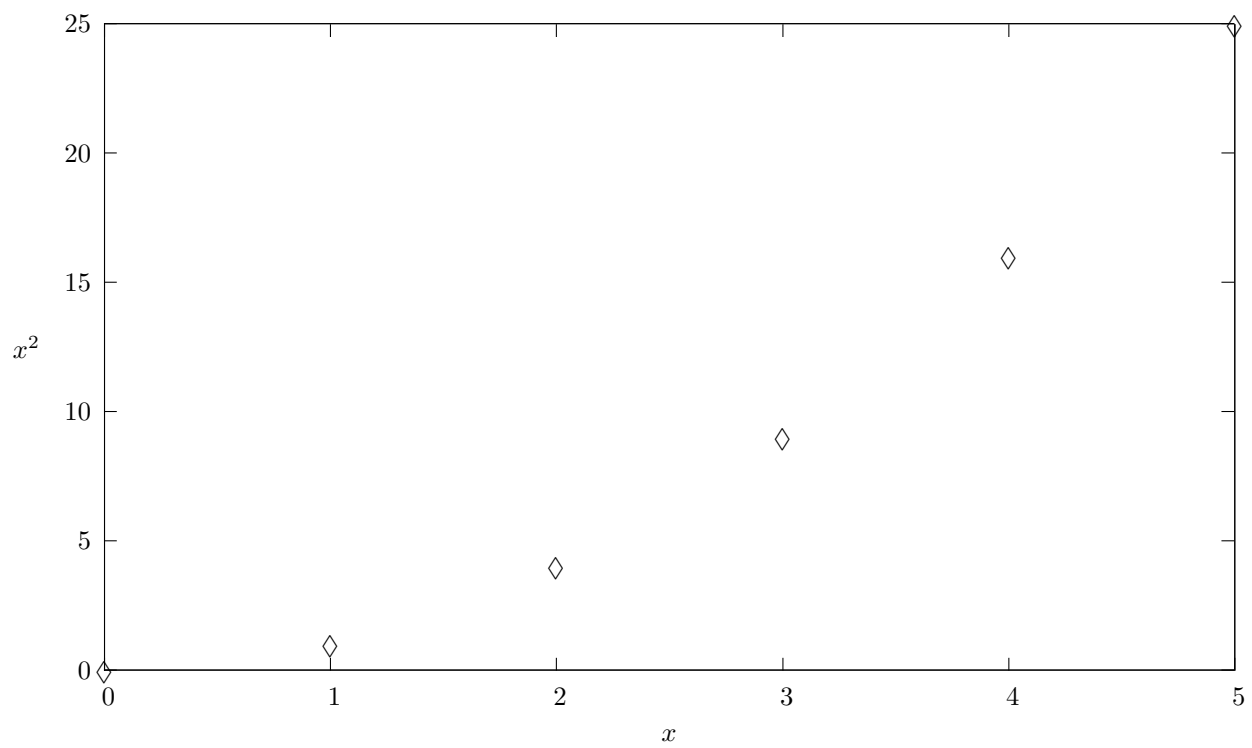
Változók használata szövegben: Helló, 5.0!

Makrók:

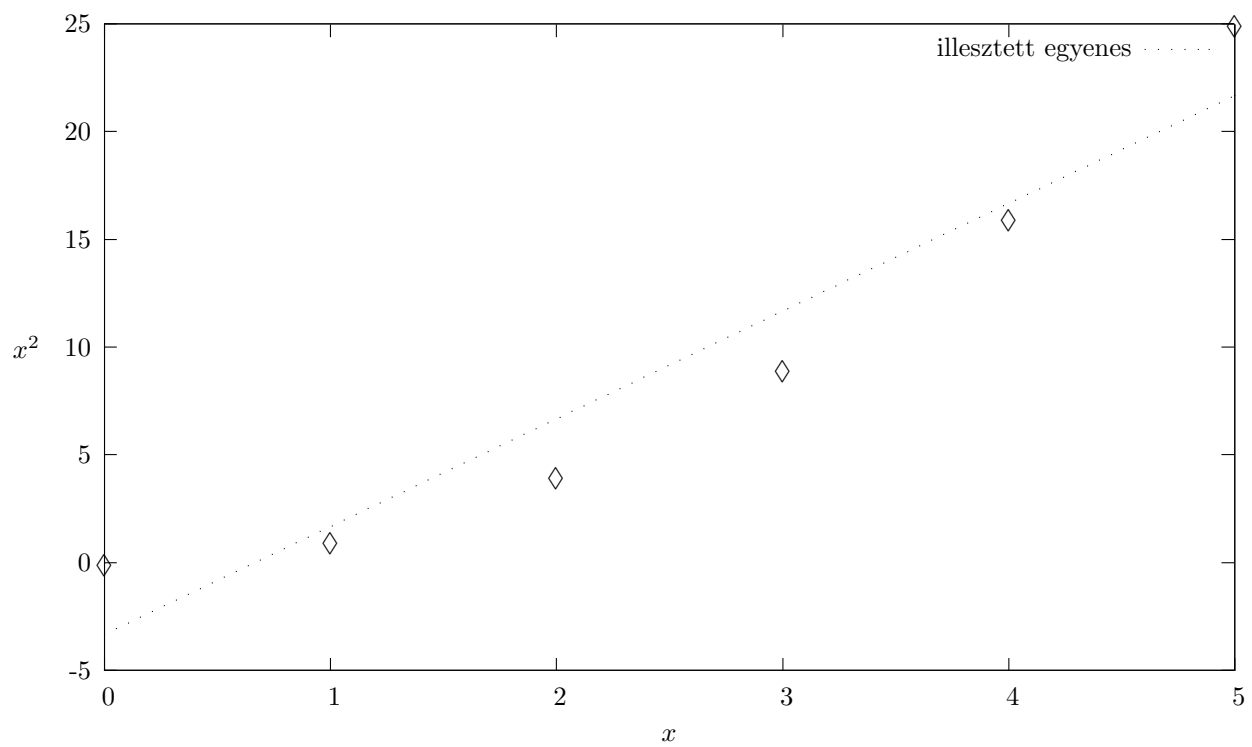
Használata: Latexben:  $k_{ij}$  a négyzetten vagy  $k_{ij}^2$ , rubyban:  $k_{ij}$  a négyzetten vagy  $k_{ij}$  a négyzetten vagy  $k_{ij}^2$  vagy  $k_{ij}^2$  vagy ennek a rövidítése:  $k_{ij}^2$ . Szóval mindenféle krix-kraxszal elő lehet hozni.

Az egész csak akkor működik rendesen, ha a makró nevében CSAK BETŰK vannak.

Figyeld meg, hogy a makró automatikusan átmegy math módba, úgyhogy mindegy hogy dollárjelek közé teszed vagy nem.  
 A :kij + valami egy bónusz egyszerűsítés, ez is magától átmegy math módba.



2. ábra. adatok



3. ábra. adatok